AIAA JOURNAL
Vol. 30, No. 6, June 1992

Domain-Decomposition Algorithm Applied to Multielement Airfoil Grids

Mark E. M. Stewart*
Princeton University, Princeton, New Jersey 08540

Multiblock grids are generated and used to solve the Euler equations for multiple-element airfoil sections. The grids are generated using an algorithm which automates several stages in the decomposition of arbitrary, two-dimensional domains into nonoverlapping, topologically rectangular regions. Because the grids are constructed so that coordinate lines are continuous across block interfaces, local accuracy estimates may be found analytically, and they indicate local second-order accuracy. The results show that multiblock grids are an effective approach to generating grids in complex two-dimensional geometries and demonstrate steps toward a fully automated domain-decomposition process.

I. Introduction

THE ability to perform aerodynamic simulations for airfoil sections and turbine blades has been limited by the techniques for generating grids or meshes in geometrically complex domains. The problem lies not only in representing the boundaries of the geometrically complex region, but also in resolving the high gradients in the domain. Examples of this problem include the suction peak of an airfoil's leading edge or the gradients near a wing tip.

Techniques that allow grids to be placed in topologically rectangular regions[1,2] form the basis of the solution to this problem, and have been extended to overlapping composite mesh systems[3,4] and multiple block grids.[5,6] Furthermore, the use of unstructured, triangular grids is an important way to represent geometrically complex domains.[7,8] The lack of rigid cell ordering in an unstructured grid allows for unlimited local refinement so that solution gradients may be properly resolved.

The current work concerns multiblock meshes or grids which are nonoverlapping, structured meshes where the coordinate lines are continuous across block interfaces. Insisting that continuous lines be continuous across interfaces simplifies the numerical procedure by eliminating the need for interpolation and replacing it with simple copying between blocks. Consequently, the resulting numerical methods show good conservation, accuracy, and stability properties. Multiblock meshes with these interface conditions can be used in parallel computations because of the simple interblock communications and ease of load balancing. These interface conditions, as currently implemented, do not have the adaptive mesh properties of unstructured or overlapping grids for resolving solution gradients.

A block decomposition must be generated for a multiblock mesh. If multiblock methods are to come into common usage, the decomposition must be automated. Several approaches have been taken in the past to generate grids in two and three dimensions that minimize the need to input data and plan the decomposition. For example, Andrews[9] uses a knowledge-based system to automate the domain decomposition in two dimensions, while Allwright[10] generates decompositions automatically from a simple block representation of an aircraft configuration. Thompson[11] details a program for generating multiblock meshes for arbitrary configurations, and Steinbrenner et al.[12] demonstrate a general, interactive system in three dimensions.

Solutions to the Euler and Navier-Stokes equations on grids of this type have been found in the past. Belk et al.[13] have demonstrated Euler solutions for a three-dimensional wing using an implicit algorithm. Flores et al.[14] have found Navier-Stokes solutions for a wing-fuselage configuration. Yadlin and Caughey[15] have shown an implicit multigrid scheme for the Euler equations on a block grid about a NACA0012 airfoil.

The objectives of the current work are to automate the domain-decomposition process and then solve the Euler equations on grids constructed for multielement airfoils. The second section briefly outlines the domain-decomposition process while the third and fourth detail how grids may then be automatically generated. Section V explains the numerical methods used to solve the Euler equations, and Sec. VI comments on their accuracy. Finally, Sec. VII demonstrates a series of results for multielement airfoils.

II. Domain-Decomposition Algorithms

The domain decompositions used to generate the current grids are found by a novel algorithm[16] which finds boundary-conforming subregions within a two-dimensional domain. In the same way that the skin of a large balloon will conform to the bounding walls when blown up in a confined space, this algorithm refines a coarse approximation to the perimeter of a region so that it conforms to any nearby neighboring boundaries without excessive stretching. The algorithm is implemented as a search tree using a small set of rules to drive a directional probe for points on nearby curves. The results are used to refine a perimeter approximation.

Some underlying ideas are demonstrated in Fig. 1 by considering the simple case of an airfoil in a box, *PQRS*, and an initial, coarse approximation to the region below the airfoil, *ACRS*. One would like to transform *ACRS* to a region which conforms to the lower surface of the airfoil. The perimeter refinement works as follows. Above the middle third of *AB*, *EF*, one may probe for the point which determines the highest flat ceiling above *EF* as in Fig. 1b. The maximum height of this probe is denoted by search depth. Limiting the depth of the probe prevents finding points on the segment *PQ* which would yield a perimeter with excessive stretching. The probe considers the curves which define the domain and finds the

Presented as Paper 90-1606 at the AIAA 21st Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Seattle, WA, June 18-20, 1990; received Feb. 11, 1991; revision received Sept. 20, 1991; accepted for publication Sept. 29, 1991. Copyright © 1992 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Graduate Student; currently Research Associate, ICOMP, MS 5-3, NASA Lewis Research Center, Cleveland, OH 44135.

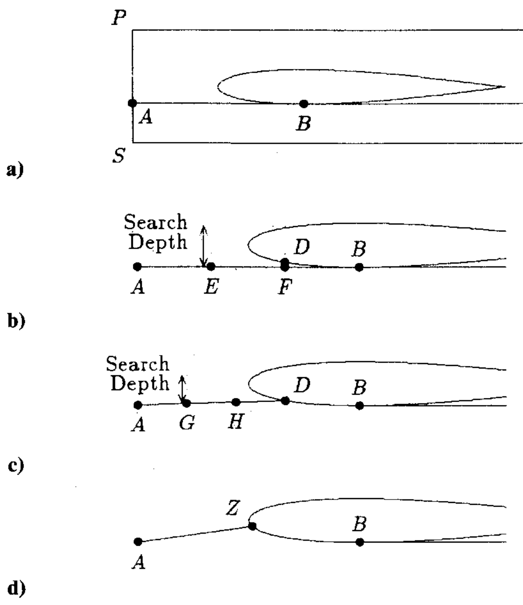

Fig. 1 Steps in finding a boundary conforming region.

point D. A small set of rules dictates that the perimeter be modified to $ADBC$, and two child segments, AD and DB, are created from AB.

The refinement of ADB involves refining AD and DB. DB is recognized as being part of a boundary-defining curve, and is not refined further, while AD is refined. Probing above the middle third of AD, GH, yields no point within search depth so the two outer segments AG and HD are considered for refinement. This failure to find a highest ceiling implies there will be a transition between bodies in the perimeter. The procedure considers refinement of successfully smaller segments as above, and eventually yields the perimeter AZB of Fig. 1d.

Because segments are created from the refinement of segments, this perimeter refinement procedure may be viewed as a depth-first tree search. The search tree is terminated when the segments are smaller than the geometrical features of the domain. Using a range of optimizations, the tree is significantly pruned and the algorithm runs satisfactorily on a Personal Iris workstation.

To find the perimeter of a region, this procedure is applied on each of the four sides of an initial rectangular approximation to a region. At the corners, the procedure is applied in two directions to remove directional considerations. The initial rectangle is found from the sole input, which is a point in the domain where one would like a boundary-conforming region.

The decomposition of a domain is achieved by finding a sequence of regions which cover the domain, and can be accomplished either automatically or manually. In automatic mode, regions are generated until the domain is covered. In manual mode, regions are generated and may be manually excluded. Also, lines which outline desired block boundaries may be included, and the algorithm will conform block boundaries to these artificial boundaries. In either mode, the interconnection information is automatically determined. Sometimes, the regions which are generated in automatic mode have features which result in grid skews and are not appropriate. Consequently, all of the current results have been generated in manual mode.

Each of the regions or blocks is topologically rectangular. Consequently, the decomposition has a topology and can have grid singularities. The two common types of singularities are three-neighbor (TN) and five-neighbor (FN) singularities.

The domain-decomposition algorithm has been applied in manual mode not only to multielement airfoils, but also to a range of two-dimensional domains. It is believed that the underlying techniques make this algorithm suitable for general two-dimensional regions.

III. Boundary Cell Allocation

The block decomposition consists of both a set of topologically rectangular blocks which cover the domain, and the boundary and interconnection information. To generate structured grids within each of these blocks, one must ensure that the cell counts on opposite sides of a block are equal. A second constraint is ensuring that coordinate lines are continuous across each of the block interfaces.

These two constraints may be automatically satisfied by a global optimization procedure. The operator is

$$\left\lfloor \frac{\text{length}(C_i \in S_J)}{\text{length}(S_J)} \times \text{cells in } S_J + P(S_J) \right\rfloor \qquad (1)$$

where length (arc) determines arc length, C_i is curve i in block side S_J, $\lfloor\ \rfloor$ is the truncate operator, and $P(\,)$ is a biasing parameter. If $P(S_J)$ is 0.5, rounding results. However, for larger values of $P(S_J)$, more than an average number of cells will be allocated; for smaller values, fewer cells will be allocated. By linking the number of mismatched cells on opposite sides of a block to this biasing parameter and iterating, a stable solution procedure results, subject to the condition that the number of cells exceeds the number of curves on each side. The second constraint can be satisfied by adding a further biasing parameter linked to the number of mismatched cells across each block interface and also by a global modification of the grid dimensions.

The dimensions required for multigrid meshes are found by refining a coarse mesh for the decomposition. The coarsest mesh is limited by the shortest block side in the decomposition. In practice, either two or three multigrid levels may be included in the meshes.

IV. Elliptic Grid Generation

To generate an initial approximation to the grid within each block of the decomposition, a simple shearing transformation based on the block boundary coordinates is used:

$$\begin{aligned} P(r,s) &= P(0,s)\cdot(1-r) + P(1,s)\cdot r \\ &\quad + P(r,0)\cdot(1-s) + P(r,1)\cdot s \end{aligned} \qquad (2)$$

Elliptic smoothing[2] with no forcing is then used to smooth the mesh. In this method, a transformation between physical space coordinates (x,y) and uniform computational space coordinates (ξ,η) is found for each topologically rectangular region by solving in $[\xi_1, \xi_2] \times [\eta_1, \eta_2]$,

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \qquad (3a)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \qquad (3b)$$

$$\alpha = x_\eta^2 + y_\eta^2, \qquad \beta = x_\xi x_\eta + y_\xi y_\eta, \qquad \gamma = x_\xi^2 + y_\xi^2 \qquad (3c)$$

With the initial interpolated grid, 20–30 Gauss-Seidel sweeps are sufficient to smooth the grid. The interfaces between blocks are also smoothed using this procedure. This ensures continuity of the transformation across block interfaces. The singularities are smoothed using averaging of neighbors. It is necessary to use more blocks in the decomposition than are required by the topology because it is difficult to initialize and form an unstretched grid if the largest possible blocks are used.

V. Numerical Methods

The two-dimensional Euler equations model inviscid, compressible, rotational flow and allow entropy and vorticity pro-

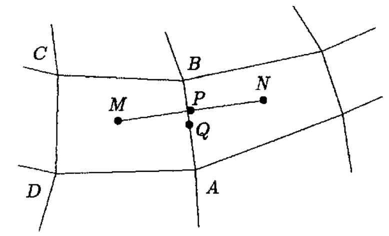

Fig. 2 Flux approximation for cell *ABCD*.

duction across shocks. They are given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} w \,\mathrm{dVol} = -\oint_{\partial\Omega} F\cdot n \,\mathrm{d}s \tag{4}$$

where $F = (f,g)$, and

$$w = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad f(w) = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uH \end{bmatrix}, \quad g(w) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ \rho vH \end{bmatrix}$$

$$P = (\gamma - 1)\left(\rho E - \frac{\rho(u^2 + v^2)}{2}\right), \qquad H = \frac{\gamma}{(\gamma - 1)}\frac{P}{\rho}$$

and $\gamma = 1.4$. The numerical approximation and solution techniques are based on the FLO52 program of Jameson et al.[17] The flux through the faces of each cell is approximated from the cell centroid-based dependent variables w. For the face AB in Fig. 2,

$$\int_{AB} f\,\mathrm{d}y - g\,\mathrm{d}x \cong \tfrac{1}{2}(f_M + f_N, g_M + g_N)\cdot(\Delta y_{AB}, -\Delta x_{AB}) \tag{5}$$

The discretized equations for each cell are advanced to a steady state by a multistage scheme,

$$w^{(1)} = w_i + \alpha_1\Delta t[Q(w_i) + D(w_i)] \tag{6a}$$

$$w^{(2)} = w_i + \alpha_2\Delta t[Q(w^{(1)}) + D(w^{(1)})] \tag{6b}$$

$$w^{(3)} = w_i + \alpha_3\Delta t[Q(w^{(2)}) + D(w^{(1)})] \tag{6c}$$

$$w^{(4)} = w_i + \alpha_4\Delta t[Q(w^{(3)}) + D(w^{(1)})] \tag{6d}$$

$$w_{i+1} = w_i + \alpha_5\Delta t[Q(w^{(4)}) + D(w^{(1)})] \tag{6e}$$

where α_i are coefficients; $Q(w)$ is the convective flux approximation for the cell, Eq. (5); and $D(w)$ is the artificial dissipation. The artificial dissipation consists of third-order dissipation, which stabilizes the time-stepping scheme, and first-order dissipation, which is switched on near shocks to capture them.

The boundary conditions at the airfoil surface ensure no convection through the surface and allow free slip. The grid around the airfoil stretches exponentially away from the airfoils into the far field and is terminated at 30 chords. At large distances, an airfoil appears to be a vortex in uniform flow. The flux integral, Eq. (5), at the far field is approximated by using this far-field velocity and matching the incoming and outgoing Riemann invariants.

The interfaces between blocks are a further type of boundary. The grid points have been allocated so that coordinate lines are continuous across these interfaces. Furthermore, the elliptic smoothing is done so that there is no loss of continuity in the transformation between the coordinate system and the domain at these boundaries. Consequently, the flux approximation of Eq. (5) and artificial dissipation of Eq. (6) are not affected by the interface since the continuation of the grid into the next block is known. This interface condition is implemented by passing data values between the grids. No interpolation is required.

A steady-state solution is sought, and to accelerate convergence to this solution, several techniques are employed. The maximum local timestep is used in each cell to accelerate convergence, and enthalpy damping is also used. Residual averaging is not used. Dramatic convergence acceleration is achieved by using a multigrid technique.[18] The number of multigrid levels is limited by the block decomposition, and in particular, the smallest block in the decomposition is the limit. Typically, only two or three multigrid levels are possible for multielement airfoil cases. The current results show that each level gives a factor of two speedup.

VI. Accuracy

Because the coordinate lines are continuous across interfaces, it is possible to determine analytic, local accuracy estimates for the numerical schemes on the nonoverlapping, structured grids which have been developed. These error estimates suggest that the truncation error in the convection terms is second-order, and it dominates the artificial dissipation.

For nonuniform grids, the Taylor expansion of the flux integral in Eq. (5) for the cell face AB in Fig. 2 is

$$\int_{AB} f\,\mathrm{d}y = f\Big|_{(x_P,y_P)}\Delta y + \frac{\partial f}{\partial x}\Big|_{(x_P,y_P)}(x_Q - x_P)\Delta y + \frac{\partial f}{\partial y}\Big|_{(x_P,y_P)}(y_Q - y_P)\Delta y + \mathcal{O}(\mathrm{Vol}^{3/2}) \tag{7}$$

where P and Q represent the midpoints of MN and AB. The use of elliptic grid smoothing to find a smooth grid, Sec. IV, is equivalent to finding a smooth transformation between the entire physical domain and the uniform coordinate system of the computational domain. Since smoothing is applied both in each block and at block interfaces, the transformation is globally smooth except at singularities. The grid coordinates are not solved to convergence by the smoothing method, Eq. (3), but the argument depends only on the transformation being C^1. Exploiting this transformation to express Eq. (7) in terms of the computational domain's coordinates, (ξ,η), one may show that $(x_Q - x_P)$ and $(y_Q - y_P)$ are second-order in the uniform coordinates. This behavior has been verified numerically. It follows that the convective flux approximation is second-order. Furthermore, the artificial dissipation is arguably dominated by the convective flux truncation error for sufficiently fine meshes where the solution is smooth.

The truncation error is not dominated solely by the second-order terms, $(x_Q - x_P)$ and $(y_Q - y_P)$, but also by the solution gradients, Eq. (7). Approximation of this truncation error indicates that gradients at the leading and trailing edges have an overwhelming effect. Hence, the order of accuracy is a secondary consideration, as are the grid distortions caused by the grid singularities.

VII. Results

Solutions to the Euler equations for several airfoils have been found using these grids. The NACA0012 airfoil was simulated and compared very well with benchmark transonic results. The Korn airfoil is a sensitive test of numerical methods and grids because it is shock free for the unique Mach number $M = 0.75$ and angle of attack $\alpha = 0$ deg. Any variation of these parameters or coding errors can result in the appearance of shocks. With 144 surface cells, the results were shock free. Details of these results are given in Ref. 19.

The Williams two-element airfoils[20] provide further test cases in low-speed flow. In this work, exact potential solutions are found analytically for flow past two circles, and transformed to a two-element airfoil using a double Karman-Trefftz transformation. A multiblock grid was generated for an airfoil using the current methods and the inner region is shown in Fig. 3. The entire grid consists of 10,688 cells in 23 blocks. The topology requires at least six blocks, and the extra blocks are needed to prevent stretching during grid smoothing. The Euler simulation took 3000 iterations with three multigrid levels and was done for $M = 0.125$, $\alpha = 0$ deg. The convergence history is shown in Fig. 4 by the rms value over the field of the density equation residual. Maintaining grid smoothness prevents more than three grid levels. However, using three multigrid levels improves convergence considerably. The rate of convergence, measured by the log residual reduction per unit work, is four times that in the single grid case. The Prandtl-Glauert similarity rule indicates that at $M = 0.125$ the C_P correction for these two simulations should be 0.8%. The predicted lift and drag coefficients are $C_L = 2.0370$ and $C_D = 0.0025$ and the exact values are $C_L = 2.0290$ and $C_D = 0.000$. The lift coefficients differ by less than 0.5%. Theoretically, there is no drag for an inviscid, subcritical flow and the numerically predicted, nonzero value is due to artificial dissipation as well as the truncation error in the flux approximation. The C_P distributions for the two elements are shown in Fig. 5, where the symbols indicate upper and lower surface numerical results and the solid line is Williams' exact, analytic solution.

A further simulation has been done for a four-element landing configuration of the A310 Airbus airfoil. The inner region of the multiblock grid for this case is shown in Fig. 6. The Euler simulation, $M = 0.3$, $\alpha = 5$ deg, took 3000 iterations using two multigrid levels; the convergence history is shown in Fig. 7. The rate of convergence, measured by the log residual reduction per unit work, is twice that of the single grid case. The grid contains 10,508 cells in 46 blocks. The topology

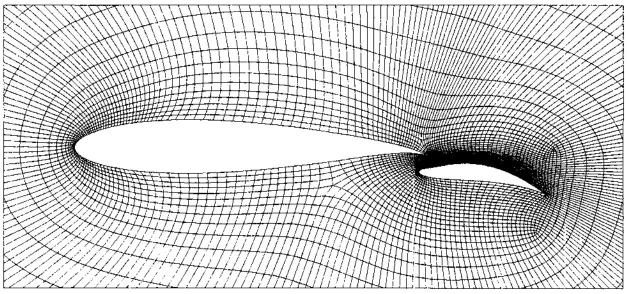

Fig. 3 Inner grid for the Williams two-element airfoil with 10-deg flap.

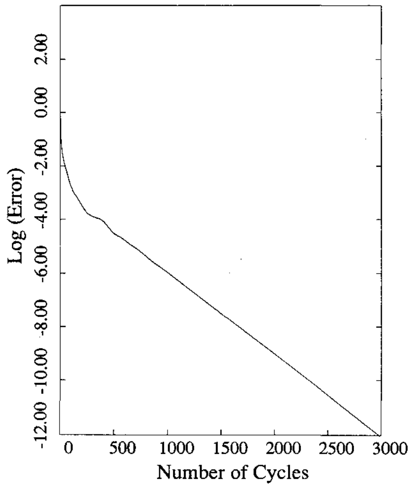

Fig. 4 Convergence history for the Williams airfoil simulation.

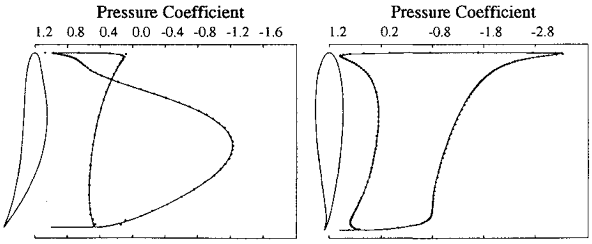

Fig. 5 Pressure distributions for the Williams airfoil simulation.

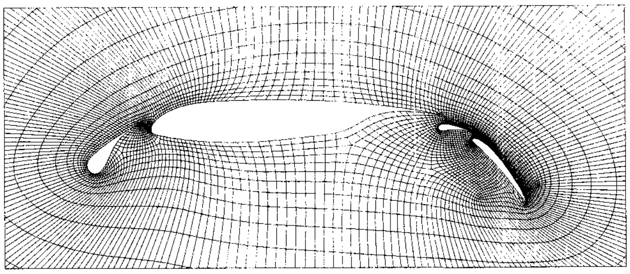

Fig. 6 Inner grid for the A310 Airbus four-element landing configuration.

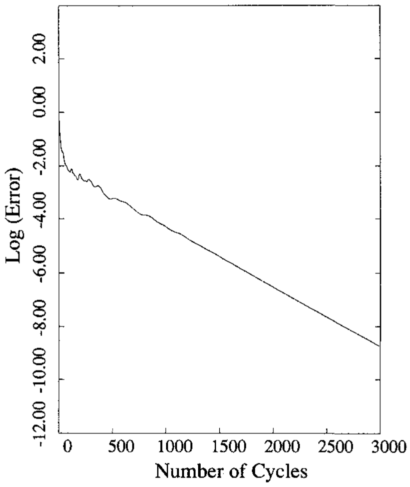

Fig. 7 Convergence history for A310 Airbus four-element simulation.

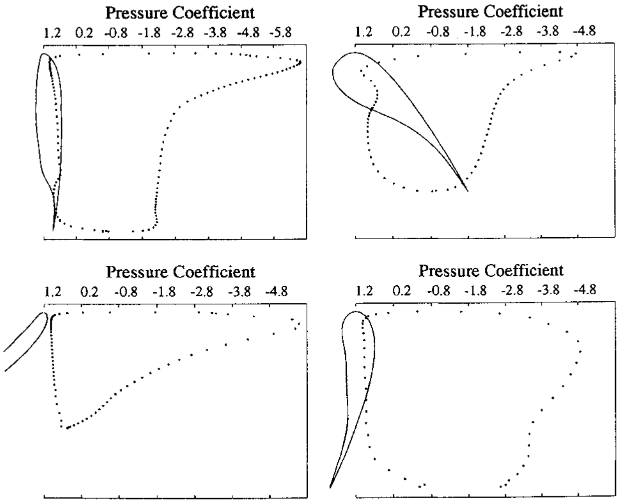

Fig. 8 Pressure distribution for the four elements of the A310 Airbus landing configuration.

requires at least 12 blocks, and the extra blocks are needed to prevent stretching during grid smoothing. The simulation predicts $C_L = 5.6608$ and $C_D = 0.0075$. The pressure distributions are shown in Fig. 8. There are no experimental results for comparison, and one must depend on the drag coefficient and convergence of the equations for verification of the solution.

VIII. Conclusions

A technique for generating nonoverlapping multiblock grids for complex two-dimensional geometries is presented and demonstrated. The Euler equations are solved for various multielement airfoils on the resulting grids. These results warrant considering extensions of the domain-decomposition algorithm to three dimensions.

Acknowledgments

This work has been done under the supervision of Antony Jameson at Princeton University. The author was supported by a postgraduate scholarship from the Natural Sciences and Engineering Research Council of Canada.

References

[1] Eiseman, P. R., "Multi-Surface Method of Coordinate Generation," *Journal of Computational Physics,* Vol. 33, No. 1, 1979, pp. 118–150.

[2] Thompson, J. F., Thames, F., and Mastin, C., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies," *Journal of Computational Physics,* Vol. 15, No. 3, 1974, pp. 299–319.

[3] Benek, J. A., Buning, P. G., and Steger, J. L., "A 3-D Chimera Grid Embedding Technique," AIAA Paper 85-1523, July 1985.

[4] Chesshire, G., and Henshaw, W., "Composite Overlapping Meshes for the Solution of Partial Differential Equations," *Journal of Computational Physics,* Vol. 90, No. 1, 1990, pp. 1–64.

[5] Lee, K. D., "3-D Transonic Flow Computations Using Grid Systems with Block Structure," AIAA Paper 81-0998, June 1981.

[6] Weatherill, N. P., and Forsey, C. R., "Grid Generation and Flow Calculations for Complex Aircraft Geometries Using a Multi-Block Scheme," AIAA Paper 84-1665, June 1984.

[7] Jameson, A., and Baker, T., "Improvements to the Aircraft Euler Method," AIAA Paper 87-0452, Jan. 1987.

[8] Lohner, R., and Parikh, P., "Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method," AIAA Paper 88-0515, Jan. 1988.

[9] Andrews, A., "Knowledge-Based Flow Field Zoning," *Proceedings of the Second International Conference on Numerical Grid Generation in CFD,* Vol. 1, Pineridge Press, Swansea, Wales, UK, 1988, pp. 13–22.

[10] Allwright, S., "Techniques in Multiblock Domain Decomposition and Surface Grid Generation," *Proceedings of the Second International Conference on Numerical Grid Generation in CFD,* Vol. 1, Pineridge Press, Swansea, Wales, UK, 1988, pp. 559–568.

[11] Thompson, J. F., "A Composite Grid Generation Code for General 3-D Regions," AIAA Paper 87-0275, Jan. 1987.

[12] Steinbrenner, J., Chawner, J., and Fouts, C., "A Structured Approach to Interactive Multiple Block Grid Generation," *AGARD FDP Specialists Meeting on Mesh Generation for Complex Three Dimensional Configurations,* Leon, Norway, May 1989.

[13] Belk, D., and Whitfield, D., "Three-Dimensional Euler Solutions on Blocked Grids Using an Implicit Two-Pass Algorithm," AIAA Paper 87-0450, Jan. 1987.

[14] Flores, J., Reznik, S., Holst, T., and Gundy, K., "Transonic Navier-Stokes Solutions for a Fighter-Like Configuration," AIAA Paper 87-0032, Jan. 1987.

[15] Yadlin, Y., and Caughey, D., "Block Multigrid Implicit Solutions of the Euler Equations of Compressible Fluid Flow," AIAA Paper 90-0106, Jan. 1990; also *AIAA Journal,* Vol. 29, No. 5, 1991, pp. 712–719.

[16] Stewart, M., "Multiblock Mesh Generation for Multiple Element Airfoils with Euler Solutions," *Computing Systems in Engineering,* Vol. 1, No. 2–4, 1990, pp. 325–331.

[17] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes," AIAA Paper 81-1259, June 1981.

[18] Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method," *Applied Mathematics and Computation,* Vol. 13, No. 3–4, 1983, pp. 327–356.

[19] Stewart, M., "Non-Overlapping Multiblock Meshes for Multi-Element Airfoils with Euler Solutions," Ph.D. Dissertation, Program in Applied and Computational Mathematics, Princeton Univ., Princeton, NJ, June 1990.

[20] Williams, B. R., "An Exact Test Case for the Plane Potential Flow About Two Adjacent Lifting Aerofoils," Royal Aeronautical Establishment, RAE RM 3717, Farnborough, England, UK, Sept. 1971.